



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/679,181	10/03/2003	Gordon Bowman	GLH 08-896329	2790
27667	7590	12/24/2009		
HAYES SOLOWAY P.C. 3450 E. SUNRISE DRIVE, SUITE 140 TUCSON, AZ 85718				
EXAMINER				
AUGUSTINE, NICHOLAS				
ART UNIT		PAPER NUMBER		
2179				
NOTIFICATION DATE		DELIVERY MODE		
12/24/2009		ELECTRONIC		

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

admin@hayes-soloway.com
smckniff@hayes-soloway.com
nsoloway@hayes-soloway.com

Office Action Summary

Application No.

10/679,181

Applicant(s)

BOWMAN ET AL.

Examiner

NICHOLAS AUGUSTINE

Art Unit

2179

Period for Reply -- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 09 October 2009.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 2, 4-9, 11-13, 16-22, 25 and 27 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 2, 4-9, 11-13, 16-22, 25 and 27 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☐ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB-08)
- 4) ☐ Interview Summary (PTO-413)
- 5) ☐ Paper No(s)/Mail Date _____
- 6) ☐ Other: _____

DETAILED ACTION

1. This action is in response to the following communications: Amendment filed: 10/09/2009. This action is made **Final**.
2. Claims 2, 4-9, 11-13, 16-22, 25 and 27 remain pending.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 4-7,11-13,16-22,25 and 27 are rejected under 35 U.S.C. 102(e) as being anticipated by Bowers, Clarke Daniel (US Pub. 2003/0203342), herein referred to as "Bowers".

As for dependent claim 4, Bowers teaches the system as claimed in claim 27, wherein the skin template is associated with the use interface control element by a reference attribute that comprises a reference to the location of a skin template file comprising the collection of presentation markup language describing the user interface control

element (par.104, 142,156,261).

As for dependent claim 5, Bowers teaches the system as claimed in claim 27, wherein the user interface control element is associated with an extensible markup language based element (par.102, 120).

As for dependent claim 6, Bowers teaches the system as claimed in claim 5, wherein the user interface control element is a parent of an extensible markup language based element (par.102, 120).

As for dependent claim 7, Bowers teaches the system as claimed in claim 5, wherein the control element is a child of an extensible markup language based element (par.102, 120).

As for dependent claim 11, Bowers teaches the system as claimed in claim 27, wherein the core attributes comprise state attributes for specifying the identification of a <state> child element of the control element (par.28).

As for dependent claim 12, Bowers teaches the system as claimed in claim 27, wherein the core attributes comprise one or more of: *an identification attribute for referencing the control element*(figures 14A-B); par.181; wherein described is storage of attributes

where the attributes are used to render the test correctly according to the attributes) ; a label attribute for associating text control; a height attribute for specifies the height of the control element (par.181; height information 613); a disabled attribute for specifying whether the control element is disabled and cannot be used; a state hover attribute for specifying the identification of a <state> child element of the control element, the state hover attribute used to override the appearance of a hover state as defined in a skin of the control element; a state focus attribute for specifying the identification of a <state> child element of the control element, the state focus attribute used to override the appearance of a focus state as defined in a skin of the control element ; a state up attribute for specifying the identification of a <state> child element of the control element, the state up attribute used to override the appearance of an up state as defined in a skin of the control element; a state down attribute for specifying the identification of a <state> child element of the control element, the state down attribute used to override the appearance of a down state as defined in a skin of the control element; a state hit attribute for specifying the identification of a <state> child element of the control element, the state hit attribute used to override the appearance of a hit state as defined in a skin of the control element; a state disabled up attribute for specifying the identification of a <state> child element of the control element, the state disabled up attribute used to override the appearance of a disabled up state as defined in the skin of the control element; and a state disabled down attribute for specifying the identification of a <state> child element of the control element, the state disabled down attribute used to override the appearance of a disabled down state as defined in a skin of the control

element (par.104; Table 1; par.116,142, 156,168,173; figure 14 and 37-44; throughout the disclosure of Bowers using at least these cited portions is explained of the process of using attributes stored in memory for rendering a graphical user interface wherein the interface is changed/modified during the course of execution of the system as disclosed by Bowers).

As for dependent claim 13, Bowers teaches the system as claimed in claim 12, wherein the set of control elements comprises one or more of: It is evident that the following: button, combo box, list box, list view, context menu, item, text box, slider, scrollbar and spin dial, are taught by Bowers through the depiction of the figures and thus only one will be analyzed in detail.

a `dsvg:button` control element for defining a control that is clicked to trigger an action, the `dsvg:button` control element comprising: a namespace following the predetermined naming convention (figure 8 and 14, par.118,183); the common attributes (par.118); other attributes comprising: a `toggle` attribute for specifying whether the button is a toggle or a sticky button; a `group` attribute for specifying the name of a group to which the button control element belongs; and a `checked` attribute for specifying whether the button control element is down/checked or up/unchecked (par.92 and 95; selection plug-in); a `skin template reference` attribute for specifying the location of a control element skin template (par.187; area storage 764), the skin template reference settable to a uniform resource index (par.268; URI used in attributes of the item); and a

customizable skin template comprising scalable vector graphics markup contained as children of a container element (par.181-187 there is described in par.181, Forms branch 600 which describes the attributes for the forms; par.182, event information 607 which describes events happening; par.183-184, items branch 650 which are element to be delivered to the form (test questions; having selection elements and the graphical means of displaying user interaction));par.185, Category branch 700 displays test categories for testing; par.186, templates branch 750 describes how to graphical display the test using template definition attributes; par.187, area storage 764 for storing the attributes of the templates; par.188, section branch 800 used for how to display linked sets of information for in response to user interaction; par.189 timer storage for attributes in navigation of timed events and par.190 for event information 856 which defines the order of event). With this outline is described a set of branches each having attributes used by the system to render a final presentation to the user; wherein the final presentation contains a graphical user interface having graphical and textual information used to test a user through interaction with selectable controls throughout the graphical user interface.

As for dependent claim 16, Bowers teaches a method of controlling user interface features of a web application, the method comprising the steps of: describing the web application using presentation markup language that has been extended to include a collection of user interface control elements (par.87-88; wherein plugins consist of user

interface control elements), the web application description including a user interface control comprising a name element for associating the user interface control of the web application with a user interface control element of the collection of user interface control elements;

searching in a document object model (DOM) of the web application, for user interface controls of the web application, the user interface controls identified by a namespace associated with user interface control elements of the collection of user interface control elements included in the extended presentation markup language (par.103,120, 220, 261; wherein the system stores user interface control elements in an object model for storage and retrieval for execution of the system to build a presentation display to the user during execution);

generating a function name associated with the user interface controls based on the namespace of the user interface control element associated with the located user interface control of the web application (par.182 the system making use of a namespace for organization of control elements to the graphical user interface being built/edited); calling user interface control instructions associated with the user interface controls through the generated function name, each of the user interface control instructions defining the behavior of the user interface control element (par.181-183); and rendering the DOM of the web application described in the extended presentation markup language, including rendering the user interface control of the web application based on at least a skin template associated with the user interface control element (par.103,120,182,191 and 238-239; wherein the system will render the final display to

the user based upon the presentation data stored within the system which consist of XML, object model objects, user interface control elements, items and helms).

As for dependent claim 17, Bowers teaches the method as claimed in claim 16, wherein the step of searching includes the steps of: traversing each node in the document object model; and determining whether the node has a name which matches a designated naming convention (par.104).

As for dependent claim 18, Bowers teaches the method as claimed in claim 16, wherein the step of calling a script includes the steps of performing a function stored in memory having the generated function name (note claim 16 above); dynamically generating a function name associated with the designated element (par.156,168,173); passing an object associated with the designated element as a parameter of the generated function(par.120,168,173); retrieving the attributes of the object; and performing a function stored in memory having the generated function name (par.125-126).

As for dependent claim 19, Bowers teaches the method as claimed in claim 18, wherein the step of generating function name includes the steps of (note claims 16 and 18 above); generating a function name comprising of the name of the designated element (par.156,168,173); assigning an object associated with the designated element as the parameter of the function (par.120,168,173); and assigning control instructions of the designated element as steps for the function to perform (par.125-126l).

As for dependent claim 20, Bowers teaches the method as claimed in claim 16, further comprising the steps of: searching for a control attribute of a user interface control element in a document object model; and calling control attribute instructions associated with the control attribute (par.183.).

As for dependent claim 21, Bowers teaches the method as claimed in claim 20, wherein the step of searching for a designated attribute comprises the steps of: searching attributes of an element in a document object model; determining whether an element attribute has a name which follows a designated naming convention (par.181, 183).

As for dependent claim 22, Bowers teaches the method as claimed in claim 21, wherein the step of calling a script includes the steps of: determining if the name of the designated attribute contains a designated prefix; generating a function name comprising of the name of the designated attribute (par.156,168,173); assigning an object associated with the designated attribute as the parameter of the function name (par.120,168,173); and assigning predetermined instructions of the designated attribute as steps for a function having the function name to perform (par.125-126).

As for dependent claim 25, Bowers teaches a method of controlling user interface features of a web application (par.87-88,120), the method comprising the steps of:

adding a behavior element as a child of a user interface control element; receiving an event which is equal to an event attribute setting in the behavior element; and calling behavior element instructions associated with the behavior element (par.96).

As for independent claim 27, Bowers teaches a system for controlling user interface features of a web application, the system comprising: a collection of user interface control elements including a user interface control element associated with a user interface control identified in the web application (par.87,88, 102), the user interface control comprising a name element for associating the user interface control of the web application with the user interface control element of the collection of user interface control elements (par.99,102), the web application described in presentation markup language that has been extended to include the collection of user interface control elements, each of the user interface control elements of the collection (par.87-88,91-92, 102,103, 120, 220) comprising:

namespace for associating the user interface control element with the user interface control of the web application, the namespace including a prefix for identifying the user interface control of the web application associated with the user interface control element as part of the extended presentation markup language (par.147148);
a set of core attributes common to all of the user interface control elements in the collection of user interface control elements (*in paragraph 87 that Bowers system is made up of various modules (parts), one of which is a test driver which is responsible for creating and putting together the interface for presentation, this test driver is a web*

application written in a deviation of XML web language (coined term XXL; "extensible exam language"). The Test driver allows for development of new plugins wherein new plugins can be customized without changing the core test driver (par.87). These plugins already present and new ones that can be added later enable the test driver to support new item types, navigation algorithms, information displays, helm types, etc... The plugins also allow for expansion of the test driver (par.88). Figure 3 shows an overview of the software architecture for the system. The test driver makes use of exam source files 130; these files store a vast amount of information crucial to the presentation of the interface (test) to the user (par.91-92; "exam source files"). Exam source files include data files 132, XXL files 134, multimedia files 136 and hypertext markup language files 138. The XXL files contain templates that determine the physical appearance of the test, plugins, and any additional data necessary to implement the test. Additional data is also stored in data files 132. In paragraph 102 and illustrated in figure 6 describes the contents of Exam source file 130, in which detailed is a collection of user interface control elements.);

a collection of skin templates (par.92-93,102), each of the skin templates associated with one of the user interface control elements through one of the core attributes of the user interface control elements (par.102), each of the skin templates describing how to display the User interface control element associated with the skin template (par.92-93,96,102), each of the skin templates described in the presentation markup language (par.27-31,92-93,96,102,110117, 134 and 137; Bowers teaches of HTML files that include any visual component of the test, such as the appearance of the test times,

questions, the appearance of the presentation on the display device, the appearance of any client specified customizations and/or the appearance of score reports (par.93). As outlined above, a function of a plugin is to change the visual style of the test and multiple plugins can be stored for use by the test driver (visual plugins; par.96). Further detailing into paragraph 102 about the contents of exam source file 120 comprises visual files that are HTML files that specify the visual presentation of the test as presented to the examine on the display device, including item files 222, presentation files 224, score report files 226 and custom look files 228. Item files 222 include HTML files that used to specify the visual component of test questions (interface control elements). The custom look files are typically HTML files with embedded script to layout. Thus the visual files change the appearance of the interface and can be customized);

a parser for parsing the web application described in the extended presentation language, the parser parsing the user interface control of the web application associated with the user interface control element into a document object model (DOM) of the web application (par.116-117; system goes through a document written in the presentation language to define symbols to objects performing a syntactic analysis of the document for presentation later to the user) and

a viewer for rendering the DOM of the web application described in the extended presentation markup language, including rendering the user interface control of the web application based on at least one of the skin templates associated with the user interface control elements, the viewer comprising a collection of user interface control

instructions, each of the user interface control instructions associated with one of the user interface control elements through a function name, the function name based on the namespace of the user interface control element, each of the user interface control instructions defining the behavior of the user interface control element to control user interface features of the web application (par.103,120,182,191 and 238-243; wherein the system will render the final display to the user based upon the presentation data stored within the system which consist of XML, object model objects, user interface control elements, items and helms).

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

6. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in

order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

7. Claims 2 and 8-9 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bowers.

As for dependent claim 2, Bowers teaches the system as claimed in claim 27, further comprising an initialization function associated with the viewer for directing the processing one or more control elements in a document object model of the extended presentation markup language defining the web application (par.120,220,261), searching for nodes identified by the namespace functions associated with the user interface control elements having names following the predetermined naming convention generating the function name based on the user interface control identifier of control elements of the nodes identified by the namespace, the function name generated following a predetermined naming convention and calling the user interface control instructions associated with the control element through the generated function name associated with the control elements, the namespace having names following a predetermined naming convention (par.116,142156,168,173) .

However, Bowers does not expressly disclose that the initialization function comprising instructions for traversing each node in a document object model. It would have been obvious to one of ordinary skill in the art at the time of the invention was made to include a initialization function comprising instructions for traversing each node in a document object model, in view of Bowers, because Bowers suggest to the skilled

artisan that in paragraphs 104 and 220; table 1; the "INavigate (navigation plug-in); which is the COM interface is used to traverse through the graphical user interface that makes up the test (GUI presented to the user) wherein the test is made up of a XML document in a DOM "document object model"; thus when the COM plug-in traverses the test it is obviously traversing a DOM which makes up the test (GUI) to access nodes in the XML document; further note par's.120,220,261).

As for dependent claim 8, Bowers teaches the system as claimed in claim 2, further comprising: a collection of control attributes for adding to the core attributes, the control attributes following the predetermined naming convention; and a collection of control attribute instructions for performing actions associated with the collection of control attributes, each instruction associated with a control attribute (figure 11 and 14; par. 150,183).

As for dependent claim 9, Bowers teaches the system as claimed in claim 8, wherein searching and calling functions associated with user interface control elements and the control attributes having names following the predetermined naming convention (par.104; table 1; par.158).

However, Bowers does not expressly disclose that the initialization function comprising instructions for traversing each node in a document object model. It would have been obvious to one of ordinary skill in the art at the time of the invention was made to include a initialization function comprising instructions for traversing each node in a

document object model, in view of Bowers, because Bowers suggest to the skilled artisan that in paragraphs 104 and 220; table 1; the "INavigate (navigation plug-in); which is the COM interface is used to traverse through the graphical user interface that makes up the test (GUI presented to the user) wherein the test is made up of a XML document in a DOM "document object model"; thus when the COM plug-in traverses the test it is obviously traversing a DOM which makes up the test (GUI) to access nodes in the XML document; further note par's.120,220,261).

(Note :) It is noted that any citation to specific, pages, columns, lines, or figures in the prior art references and any interpretation of the references should not be considered to be limiting in any way. A reference is relevant for all it contains and may be relied upon for all that it would have reasonably suggested to one having ordinary skill in the art. In re Heck, 699 F.2d 1331, 1332-33, 216 USPQ 1038, 1039 (Fed. Cir. 1983) (quoting In re Lemelson, 397 F.2d 1006,1009, 158 USPQ 275, 277 (CCPA 1968)).

Response to Arguments

Applicant's arguments filed 10/09/2009 have been fully considered but they are not persuasive.

After careful review of the amended claims (given the broadest reasonable interpretation) and the remarks provided by the Applicant along with the cited reference(s) the Examiner does not agree with the Applicant for at least the reasons provided below.

A1. Applicant argues that Bowers does not teach " *a collection of user interface control elements including a user interface control element associated with a*

user interface control identified in the web application, the user interface control comprising a name element for associating the user interface control of the web application with the user interface control element of the collection of user interface control elements, the web application described in presentation markup language that has been extended to include the collection of user interface control elements, each of the user interface control elements of the collection;

a collection of skin templates, each of the skin templates associated with one of the user interface control elements through one of the core attributes of the user interface control elements, each of the skin templates describing how to display the User interface control element associated with the skin template, each of the skin templates described in the presentation markup language” in the claim language, particularly claim 27.

R1. Examiner does not agree, Bowers teaches in paragraph 87 that Bowers system is made up of various modules (parts), one of which is a test driver which is responsible for creating and putting together the interface for presentation, this test driver is a web application written in a deviation of XML web language (coined term XXL; “extensible exam language”). The Test driver allows for development of new plugins wherein new plugins can be customized without changing the core test driver (par.87). These plugins already present and new ones that can be added later enable the test driver to support new item types, navigation algorithms, information displays, helm types, etc... The plugins also allow for expansion of the test driver (par.88). Figure 3 shows an overview

of the software architecture for the system. The test driver makes use of exam source files 130; these files store a vast amount of information crucial to the presentation of the interface (test) to the user (par.91-92; "exam source files"). Exam source files include data files 132, XXL files 134, multimedia files 136 and hypertext markup language files 138. The XXL files contain templates that determine the physical appearance of the test, plugins, and any additional data necessary to implement the test. Additional data is also stored in data files 132. In paragraph 102 and illustrated in figure 6 describes the contents of Exam source file 130, in which detailed is a collection of user interface control elements. Thus Bowers provides support for the limitation *"a collection of user interface control elements including a user interface control element associated with a user interface control identified in the web application" and "the web application described in presentation markup language that has been extended to include the collection of user interface control elements, each of the user interface control elements of the collection"*. Further in paragraph 99, Bowers describes how collection of interface items are linked (object-linking and embedding or OLE) and stored thus providing support for the limitation; *" the user interface control comprising a name element for associating the user interface control of the web application with the user interface control element of the collection of user interface control elements"*.

Further in addition to what was discussed above, Bowers teaches of HTML files that include any visual component of the test, such as the appearance of the test times, questions, the appearance of the presentation on the display device, the appearance of

any client specified customizations and/or the appearance of score reports (par.93). As outlined above, a function of a plugin is to change the visual style of the test and multiple plugins can be stored for use by the test driver (visual plugins; par.96). Further detailing into paragraph 102 about the contents of exam source file 120 comprises visual files that are HTML files that specify the visual presentation of the test as presented to the examine on the display device, including item files 222, presentation files 224, score report files 226 and custom look files 228. Item files 222 include HTML files that used to specify the visual component of test questions (interface control elements). The custom look files are typically HTML files with embedded script to layout. Thus the visual files change the appearance of the interface and can be customized, therefore Bowers teaches " *a collection of skin templates; each of the skin templates associated with one of the user interface control elements through one of the core attributes of the user interface control elements; each of the skin templates describing how to display the user interface control element associated with the skin template and each of the skin templates described in the presentation markup language*". Further Bowers teaches using classes for storing templates (par.110, 117, 134, 137). General Template information can be found in paragraphs 27-31, 92 and 102.

A2. Applicant argues that Bowers can not be obvious to teach the argument of A1 above.

R2. Examiner showed how Bowers anticipates the claim language in R1 above.

Conclusion

THIS ACTION IS MADE FINAL. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Inquires

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Nicholas Augustine whose telephone number is 571-270-1056 and fax is 571-270-2056. The examiner can normally be reached on Monday - Friday: 9:30am- 5:00pm Eastern.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on 571-272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Nicholas Augustine/
Examiner
Art Unit 2179
December 16, 2009

/Ba Huynh/
Primary Examiner, Art Unit 2179